Rapid Communications

Learning phase transitions from dynamics

Evert van Nieuwenburg,[1] Eyal Bairey,[2] and Gil Refael[1]
[1]*Institute for Quantum Information and Matter, Caltech, Pasadena, California 91125, USA*
[2]*Physics Department, Technion, Haifa 3200003, Israel*

(Received 23 December 2017; revised manuscript received 10 April 2018; published 9 August 2018)

We propose the use of recurrent neural networks for classifying phases of matter based on the dynamics of experimentally accessible observables. We demonstrate this approach by training recurrent networks on the magnetization traces of two distinct models of one-dimensional disordered and interacting spin chains. The obtained phase diagram for a well-studied model of the many-body localization transition shows excellent agreement with previously known results obtained from time-independent entanglement spectra. For a periodically driven model featuring an inherently dynamical time-crystalline phase, the phase diagram that our network traces coincides with an order parameter for its expected phases.

DOI: 10.1103/PhysRevB.98.060301

Introduction. Machine learning is emerging as a novel tool for identifying phases of matter [1–15]. At its core, this problem can be cast as a classification problem in which data obtained from physical systems are assigned a class (i.e., a phase) using machine learning methods. This approach has enabled the autonomous detection of order parameters [2,5,6], phase transitions [1,3], and entire phase diagrams [4,7,16,17]. Simultaneous research effort at the interface between machine learning and many-body physics has focused on the use of neural networks for efficient representations of quantum wave functions [18–26], drawing a parallel between deep networks and the renormalization group [27–29]. Overall, these studies exemplify the power of machine learning for extracting information from physical data without detailed physical input. In particular, it shows the potential for identifying novel phases through the automatic processing of large-scale data, possibly identifying features that may have been missed before.

So far, these methods have relied only on static properties of the underlying physical systems, such as raw state configurations sampled from Monte Carlo simulations [1,15] or entanglement spectra obtained using exact diagonalization [3,11,17]. However, dynamics of physical observables are often more accessible experimentally. It is therefore important to investigate how these methods can be adapted for studying phase transitions from dynamical data.

Here, we suggest a machine learning approach to distinguish between phases based on the dynamics of measurable quantities. Specifically, we introduce the use of recurrent neural networks (RNNs), designed for processing sequential data such as time traces. This approach does not rely on thermal equilibrium, and applies very naturally to time-dependent systems. It is therefore particularly suited for the identification of dynamical as well as Floquet phases [30–39].

We first test our method on a system with two inherently different dynamical behaviors, namely, a one-dimensional (1D) system with a many-body localization (MBL) transition [40–43]. Machine learning methods applied on entanglement spectra of eigenstates were used to obtain a phase diagram of the same model [11], as well as on a slightly different model featuring two distinct MBL phases [17]. Here, we insist on using only experimentally relevant (i.e., measurable) quantities such as the magnetization of individual spins. We find that the network succeeds at distinguishing between the ergodic and localized phases of this model, recovering phase boundaries similar to those obtained by previous methods.

We then apply our method to a periodically driven model, featuring among its three phases one which is unique to the time-dependent setting, namely, a time crystal [44–50]. Indeed, the method distinguishes between the time-crystalline, Floquet-ergodic, and Floquet-MBL [51–53] phases of this model.

In the following, we first introduce the essentials of recurrent neural networks. We refer the reader to Ref. [54] for an extensive introduction to the nonrecurrent feed-forward neural network. After we have introduced the network essentials, we outline the procedure we refer to as "blanking" for training the network on a set of physics data. This framework is independent of the underlying model, and serves as the main supervised learning scheme in our work. Next, we turn to introducing the models and the results mentioned earlier, and conclude with a critical evaluation of the obtained results.

Recurrent networks. Because we wish to be able to capture non-equal-time correlations in the magnetization traces, we choose to train a recurrent neural network (RNN) to distinguish dynamical regimes. A recurrent neural network is a neural network in which one or multiple outputs are fed back into the network as inputs, as illustrated in Fig. 1. Such a recurrence creates a feedback loop that allows information that was fed into the network to persist in a self-consistent manner. This is ideal for analyzing sequences in which the value at a particular point of that sequence may depend on the previous entries. Consequently, RNNs are well suited for dealing with sequential data or other types of data for which a kind of "memory" or temporal dependence is beneficial.

It is particularly useful to introduce the idea of "unrolling" a recurrent part of a network. In Fig. 1 we show (a subsection of) a neural network $\mathcal{N}$ with inputs $x(t)$ and outputs $y(t)$, the latter being fed back into the inputs. We think of t here as a discrete parameter, such that inputs and outputs are computed at time steps t, $t+1$, etc. The feedback should

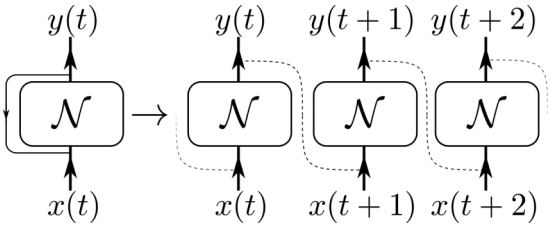

FIG. 1. Unrolling a recurrent network. On the left, (a subpart of) a neural network $\mathcal{N}$ is shown with output feeding back into the input, making it into a recurrent neural network. On the right, the unrolled version of the same network is shown, detailing that the output at step t is fed back as an input for time step $t+1$. The recurrent connections have their own weights that are optimized during training.

now be understood such that at time step t, the network receives both $x(t)$ and $y(t-1)$ as its inputs, and produces $y(t)$ from them via an intermediate step. This is most easily visualized by the unrolled network shown in Fig. 1. Namely, the network keeps track of an internal state $h(t)$, which is updated according to $h(t) = f(h(t-1), x(t))$. The function f represents the free parameters that we wish to learn by training the network. Given $h(t)$, the output $y(t) = g(h(t))$ is computed via another learnable function. The training of such a network is done in a supervised manner identical to the standard feed-forward networks, except that it can be thought of as done "unrolled layer" by "unrolled layer." There are various choices for the functions f and g introduced above, and we use so-called long short term memory networks (LSTMs) [55]. We expect that the recurrence allows the network to build a better model governing the dynamics, which helps it in the task of classifying the inputs.

Blanking. Since training the recurrent network requires labeled data (it is a supervised method), we use physics intuition to label the data only in the extremities of the phase space we consider, i.e., in the limits where we are confident about the physics of the system. The network is trained only in these regimes, and hence instead of the network seeing all the data, we effectively "blank out" outside of the known limits. This blanking tests the network's ability to extract the underlying essential model of the data from these limits, and apply it to unseen data as a form of generalization. Care must be taken that one supplies the network with enough and representative data such that a consistent model can, at least in principle, be extracted. As an important check we have tested that the predictions of the network are insensitive to adding slightly more or slightly less labeled data at the extremities (i.e., by shrinking or enlarging the blanked out region), that the network's confidence is correlated with its accuracy [56], and that the network assigns a confused output to phases it had not encountered during training [57].

Additionally, one must check for and prevent the possibility of the network learning examples by heart (i.e., overfitting). We will employ dropout [58] and weight decay (l_2 regularization) to do so. We remark that empirically for models with disorder the many realizations and their variety even for a given disorder strength seem to already build in an inherent robustness against overfitting. The actual training of the network is done by minimizing the cross entropy using the Adam optimizer [59]. Additionally, we remark that the usual test-set validation cannot be performed in the blanked region, since the network is not trained there.

Given the number n of regions in which we know the physics (i.e., the number of expected phases), our networks are constructed with a softmax output layer with n neurons. Thus, the networks take a sequence of magnetizations and output a probability distribution $\mathbf{p} = (p_1, \ldots, p_n)$ over the n phases. This distribution describes the probability that the network assigns for the input sequence to belong to each of the phases $1, \ldots, n$. We then measure the confusion (uncertainty) of the network by examining the reduced distribution on the two most likely phases. Namely, assuming the probabilities are ordered by decreasing magnitudes ($p_1 \geqslant p_2 \geqslant \cdots$), we define the confusion as $C = -\log_2[p_1/(p_1 + p_2)]$. The confusion C vanishes when the network confidently predicts a specific phase ($p_1 = 1$, $p_2 = 0$), and it takes the maximal value of unity whenever the network cannot decide between two or more phases ($p_1 = p_2$). Whenever the network changes its prediction from one phase to another at a certain value of an underlying parameter, the peak in C surrounding this value can hence indicate the corresponding transition region.

MBL transition. We consider the random-field Heisenberg model [60],

$$H = \sum_i J\, \boldsymbol{S}_i \cdot \boldsymbol{S}_{i+1} + w_i S_i^z. \tag{1}$$

The length of the chain is given by L, and the on-site disorders w_i are drawn independently and uniformly from the interval $[-W, W]$. This Hamiltonian exhibits a transition between a delocalized and a many-body localized state at a critical disorder strength that depends on the energy density of the state under consideration [60–63]. The dynamics of initial product states of spin polarization differs substantially between the two phases: While spins in the many-body localized phase retain a long-term correlation with their initial configuration, in the delocalized phase this correlation is lost over time as expected from an ergodic system [64–67]. In what follows we will be considering the dynamics of initial states that evolve in time under the Hamiltonian of Eq. (1), by performing exact time evolution on systems of size $L = 20$.

For the purpose of obtaining a phase diagram, we probe the dynamics at various energy densities. Similarly to Ref. [61], we measure the energy density by a parameter ε interpolating between the minimal and maximal eigenenergies E_0, $E_{\max}$ of each disorder realization. For each disorder realization we calculate E_0, $E_{\max}$, and pick the product state in the S_z basis ($|\uparrow, \uparrow, \downarrow, \uparrow, \ldots\rangle$, etc.) whose energy expectation value is closest to $E = E_0 + \varepsilon(E_{\max} - E_0)$. We numerically evolve this initial state in time and measure $\langle \sigma_i^z \rangle(t)$ for each of the spins.

The input to our networks therefore consists of these magnetization time traces from $t = 0$ to $t = 500$, which we sample at 50 equally spaced points, and hence is of shape $(L, 50)$ for each disorder realization. At all energy densities considered we assume that the weak disorder regime ($W \leqslant 0.5$) is ergodic while the strong disorder regime ($W \geqslant 7.5$) is many-body localized. We therefore train the network on magnetization traces from these two extreme regimes. At low disorder these traces are labeled by a label $\mathbf{p} = (1, 0)$, and at high disorder the label assigned is $\mathbf{p} = (0, 1)$. Any data for disorder strengths in the interval $W \in [0.5, 7.5]$ are therefore blanked out.

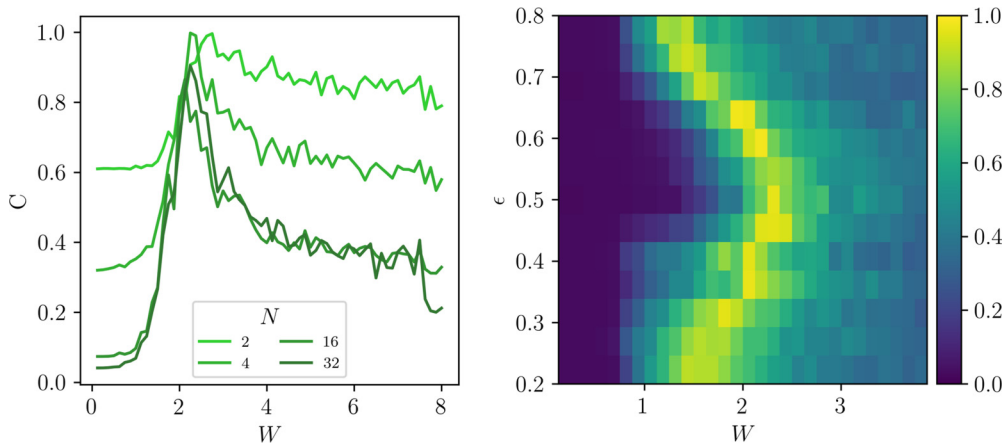

FIG. 2. Detecting the MBL transition in the random-field Heisenberg model (1). In the left panel, we show the dependence of the network's confusion C on the number of LSTM neurons N for $\varepsilon = 0.5$ and a fixed set of parameters: dropout 0.2, $l_2 = 0.01$, batch size of 64 and 25 training epochs. The right panel shows the resulting phase diagram (the color bar represents the confusion C) in the ε vs W plane, obtained with $N = 32$ and averaged over ten retrainings.

We fix the network architecture to have a single hidden layer of N LSTM neurons with a dropout rate of 0.2 and l_2 regularization of 0.01, followed by a softmax layer to output a probability of the input being ergodic or nonergodic. In the results below, we have retrained the network $k = 10$ times with identical parameters but different initial conditions. The results are averaged over these training cases.

We analyze the dependence of the output on the number N of LSTM units in the left panel of Fig. 2, and find that with 32 neurons we are able to converge the results for fixed batch size 64 and 25 epochs. This training was done on the $\varepsilon = 0.5$ data, and uses the confidence enhancement introduced in Ref. [11]. In order to gain a better understanding of what the LSTM neurons are doing, we analyze the case of a single LSTM neuron trained on a single-spin subsystem in the Supplemental Material [57]. To obtain the phase diagram, we repeat the training process over the two-dimensional parameter space of energy density (13 values equally spaced between $\varepsilon = 0.2$ and $\varepsilon = 0.8$) and disorder strength (64 values equally spaced between $W = 0.125$ and $W = 8$), with 50 disorder realizations for each point. The obtained phase diagram is shown in the right panel of Fig. 2, and shows good agreement with the phase diagram obtained from static entanglement spectra in Ref. [11].

Time crystals. Next, we consider the following binary Floquet Hamiltonian acting on a one-dimensional spin-1/2 chain,

$$H = \begin{cases} (g-\epsilon)\sum_i \sigma_i^x, & 0 < t < T_1, \\ \sum_i J_i \sigma_i^z \sigma_{i+1}^z + B_i^z \sigma_i^z, & T_1 < t < T_2, \end{cases} \tag{2}$$

where J_i, B_i are random variables distributed independently and uniformly in the interval $[0, 0.5]$, g is fixed to $\pi/2$, and $T_1 + T_2 = T$. This is a slight variation of the model studied in Ref. [46], where we took a different distribution for the bond terms J_i. As we explain below, this model features an inherently dynamical phase that cannot be studied in a static setting.

We are interested in the effect of the driving parameter ϵ on the resulting phase of the system. A guideline for the phases is provided through the long-time imbalance $\mathcal{I}(t)$ defined as

$$\mathcal{I}(t) = \frac{1}{L}\mathbf{m}(t) \cdot \mathbf{m}(0), \tag{3}$$

where the ith component of $\mathbf{m}(t)$ is the expectation value of σ_i^z at time t. This definition of the imbalance is the direct generalization of that typically used when the initial state is only taken to be one with a charge-density-wave ordering [64–67].

The long-time imbalance shows three distinct behaviors as a function of the driving parameter ϵ (orange line in

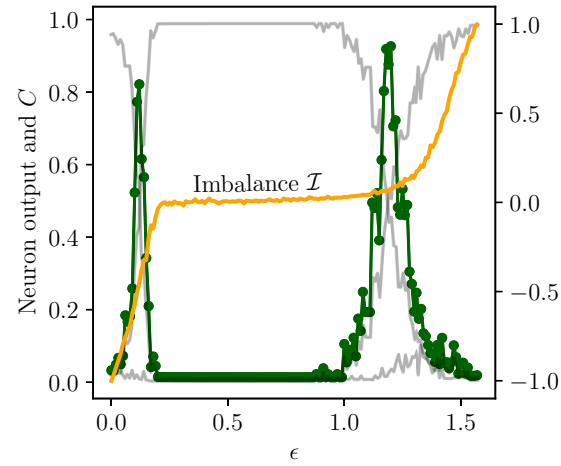

FIG. 3. A recurrent neural network distinguishes between three dynamical phases of a time-dependent model, after being trained on example curves $\mathbf{m}(t)$ at $\epsilon = 0$, 0.7, and $\pi/2$. The gray curves show the outputs of the three neurons assigned to recognize each of the three phases (time-crystalline, Floquet-ergodic, and Floquet-MBL). In green (with dots) the confusion C of the network is shown, indicating two transition points between these phases. In orange we show the long-time imbalance $\mathcal{I}(t)$ measured at an odd driving period, taking a negative value in the time-crystalline phase, a vanishing value in the Floquet-ergodic phase, and a positive value in the Floquet-MBL phase. The phase boundaries extracted by the network are consistent with, and seem sharper than, the imbalance.

Fig. 3). If $\epsilon = g = \pi/2$, the drive term is just an identity operator and the system is governed by a many-body localized Hamiltonian. Subsequently, for ϵ sufficiently close to $\pi/2$, the imbalance retains a value close to its initial one, indicating a trivial Floquet-MBL phase. For intermediate values of ϵ, the long-time imbalance vanishes, indicating a transition to a Floquet-ergodic phase. Interestingly, below a critical value of ϵ, the long-term imbalance retains a value that is close to its initial one in magnitude, but flips sign every driving period. In this regime the system's response is periodic in $2T$ rather than T, leading to the nomenclature "time crystal."

We proceed with training a RNN on time traces of $\mathbf{m}(t)$ identically to the case of the previously discussed MBL system, apart from having three regions in phase space where we train the network instead of two. Namely, for ϵ close to 0 we assign the time-crystalline label, for $\epsilon \approx 0.7$ we assign the Floquet-ergodic label, and for $\epsilon = \pi/2$ we assign the Floquet-MBL label. We again use 32 LSTM units, dropout 0.2, and $l_2 = 0.01$ with Adam optimization. When evaluated on a data set with many more ϵ available, the resulting 1D phase diagram is shown in Fig. 3 (green and gray lines).

Discussion and prospects. The main point considered in this Rapid Communication was the study of *dynamics* using machine learning methods, and doing so using *experimentally* available measurements. We employed recurrent neural networks, rather than their nonrecurrent variants. There are multiple motivations, apart from the input being sequences, for using such an approach over more common nonrecurrent feed-forward networks. First, since the data are fed into the network one time step at a time, the number of network parameters does not scale with the number of time steps. This also means that the same recurrent network can be easily trained on various lengths of data. In contrast, a regular feed-forward network would need to be input with all of the data at once, leading to a large initial input layer compatible with a fixed input length. We have studied whether the use of recurrent neurons provides a more direct way of extracting what feature of the data the neurons use to output their guess. By training one or multiple LSTM units on single magnetization curves, it is possible to identify neuron behavior [57]. We speculate that it might be possible to extract from these results a "dynamical order parameter," which takes the full magnetization traces into account.

Using the networks, we constructed dynamical phase diagrams for the MBL transition and a driven model featuring a time-crystalline phase, thereby circumventing the need to manually construct a threshold criterion or dynamical order parameter for locating the phase boundary. Rather, such a threshold was automatically determined from the data. By considering the LSTM neuron outputs for the single spin case [57], we were able to gain some understanding of the behavior that the network latches onto.

We emphasize that obtaining a phase boundary from data can hence only be as accurate as the available data. The boundary we obtain for the MBL transition is at a slightly lower disorder strength than that of the exact diagonalization results in Ref. [61], but agrees well with that obtained using the machine learned entanglement spectra of Ref. [11]. The alternative of finding a non-machine-learned proxy to serve as an indicator, such as the imbalance for the MBL transition [57], can be ambiguous. If sufficient data are available, we expect the consistency of that data to be the judge of where the transition happens. It may be possible to use the same criterion in a feedback system between a machine learning algorithm and experiments, where measurements performed on the experiment are chosen to improve the phase boundary.

For the MBL transition, in particular, we mention that a more detailed investigation should also take into account the possibility of a Griffiths phase, possibly showing up as a region where the network prediction is increasingly uncertain as system size increases. Such a finite-size scaling can indeed be successfully attempted using machine learned data [1,15], and provides a useful and interesting alternative for locating a phase boundary.

Being able to train recurrent neural networks on time traces of data poses the question of whether such methods can be used to enhance the prediction of dynamics, i.e., in numerical time evolution simulations. Such questions are being actively addressed in order to provide accurate control over, e.g., single qubits in decohering environments and noisy measurements [68].

Acknowledgments. E.v.N. gratefully acknowledges financial support from the Swiss National Science Foundation through Grant No. P2EZP2-172185. E.v.N. also acknowledges fruitful discussions with Manuel Endres. E.B. is grateful to Netanel Lindner for his support and acknowledges financial support from the European Research Council (ERC) under the European Union Horizon 2020 Research and Innovation Programme (Grant Agreement No. 639172). G.R. is grateful to the the NSF for funding through Grant No. DMR-1040435 as well as the Packard Foundation. We are grateful for support from the IQIM, an NSF physics frontier center funded in part by the Moore Foundation. The authors used the TENSORFLOW [69] backend for KERAS [70].

E.v.N. and E.B. contributed equally to this work.

[1] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017).
[2] L. Wang, Phys. Rev. B **94**, 195105 (2016).
[3] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Nat. Phys. **13**, 435 (2017).
[4] P. Broecker, F. Assaad, and S. Trebst, arXiv:1707.00663.
[5] S. J. Wetzel and M. Scherzer, Phys. Rev. B **96**, 184410 (2017).
[6] S. J. Wetzel, Phys. Rev. E **96**, 022140 (2017).
[7] Y.-H. Liu and E. P. L. van Nieuwenburg, Phys. Rev. Lett. **120**, 176401 (2018).
[8] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Phys. Rev. X **7**, 031038 (2017).
[9] K. Ch'ng, N. Vazquez, and E. Khatami, Phys. Rev. E **97**, 013306 (2018).
[10] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Sci. Rep. **7**, 8823 (2017).
[11] F. Schindler, N. Regnault, and T. Neupert, Phys. Rev. B **95**, 245134 (2017).
[12] T. Ohtsuki and T. Ohtsuki, J. Phys. Soc. Jpn. **85**, 123706 (2016).

[13] L.-F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, Phys. Rev. B **90**, 155136 (2014).
[14] L.-F. Arsenault, O. A. von Lilienfeld, and A. J. Millis, arXiv:1506.08858.
[15] M. J. Beach, A. Golubeva, and R. G. Melko, Phys. Rev. B **97**, 045207 (2018).
[16] N. Yoshioka, Y. Akagi, and H. Katsura, Phys. Rev. B **97**, 205110 (2018).
[17] J. Venderley, V. Khemani, and E.-A. Kim, Phys. Rev. Lett. **120**, 257204 (2018).
[18] G. Carleo and M. Troyer, Science **355**, 602 (2017).
[19] M. Schmitt and M. Heyl, SciPost Phys. **4**, 013 (2018).
[20] Z. Cai and J. Liu, Phys. Rev. B **97**, 035116 (2018).
[21] Y. Huang and J. E. Moore, arXiv:1701.06246.
[22] D.-L. Deng, X. Li, and S. D. Sarma, Phys. Rev. B **96**, 195145 (2017).
[23] Y. Nomura, A. Darmawan, Y. Yamaji, and M. Imada, Phys. Rev. B **96**, 205152 (2017).
[24] D.-L. Deng, X. Li, and S. D. Sarma, Phys. Rev. X **7**, 021021 (2017).
[25] X. Gao and L.-M. Duan, Nat. Commun. **8**, 662 (2017).
[26] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Nat. Phys. **14**, 447 (2018).
[27] P. Mehta and D. J. Schwab, arXiv:1410.3831.
[28] M. Koch-Janusz and Z. Ringel, Nat. Phys. **14**, 578 (2018).
[29] S.-H. Li and L. Wang, arXiv:1802.02840.
[30] R. Moessner and S. L. Sondhi, Nat. Phys. **13**, 424 (2017).
[31] N. H. Lindner, G. Refael, and V. Galitski, Nat. Phys. **7**, 490 (2011).
[32] P. Titum, E. Berg, M. S. Rudner, G. Refael, and N. H. Lindner, Phys. Rev. X **6**, 021013 (2016).
[33] F. Nathan, M. S. Rudner, N. H. Lindner, E. Berg, and G. Refael, Phys. Rev. Lett. **119**, 186801 (2017).
[34] D. V. Else, B. Bauer, and C. Nayak, Phys. Rev. X **7**, 011026 (2017).
[35] M. Heyl, Rep. Prog. Phys. **81**, 054001 (2018).
[36] C. W. von Keyserlingk and S. L. Sondhi, Phys. Rev. B **93**, 245145 (2016).
[37] D. V. Else and C. Nayak, Phys. Rev. B **93**, 201103 (2016).
[38] H. C. Po, L. Fidkowski, T. Morimoto, A. C. Potter, and A. Vishwanath, Phys. Rev. X **6**, 041070 (2016).
[39] A. C. Potter, T. Morimoto, and A. Vishwanath, Phys. Rev. X **6**, 041001 (2016).
[40] D. Basko, I. Aleiner, and B. Altshuler, Ann. Phys. **321**, 1126 (2006).
[41] V. Oganesyan and D. A. Huse, Phys. Rev. B **75**, 155111 (2007).
[42] R. Nandkishore and D. A. Huse, Annu. Rev. Condens. Matter Phys. **6**, 15 (2015).
[43] D. A. Abanin and Z. Papić, Ann. Phys. **529**, 1700169 (2017).
[44] D. V. Else, B. Bauer, and C. Nayak, Phys. Rev. Lett. **117**, 090402 (2016).
[45] V. Khemani, A. Lazarides, R. Moessner, and S. L. Sondhi, Phys. Rev. Lett. **116**, 250401 (2016).
[46] N. Y. Yao, A. C. Potter, I.-D. Potirniche, and A. Vishwanath, Phys. Rev. Lett. **118**, 030401 (2017).
[47] F. Wilczek, Phys. Rev. Lett. **109**, 160401 (2012).
[48] S. Choi, J. Choi, R. Landig, G. Kucsko, H. Zhou, J. Isoya, F. Jelezko, S. Onoda, H. Sumiya, V. Khemani *et al.*, Nature (London) **543**, 221 (2017).
[49] J. Zhang, P. W. Hess, A. Kyprianidis, P. Becker, A. Lee, J. Smith, G. Pagano, I.-D. Potirniche, A. C. Potter, A. Vishwanath *et al.*, Nature (London) **543**, 217 (2017).
[50] C. W. von Keyserlingk, V. Khemani, and S. L. Sondhi, Phys. Rev. B **94**, 085112 (2016).
[51] P. Ponte, Z. Papić, F. Huveneers, and D. A. Abanin, Phys. Rev. Lett. **114**, 140401 (2015).
[52] A. Lazarides, A. Das, and R. Moessner, Phys. Rev. Lett. **115**, 030402 (2015).
[53] P. Bordia, H. Lüschen, U. Schneider, M. Knap, and I. Bloch, Nat. Phys. **13**, 460 (2017).
[54] M. Nielsen, *Neural Networks and Deep Learning* (Determination Press, 2015).
[55] S. Hochreiter and J. Schmidhuber, Neural Comput. **9**, 1735 (1997).
[56] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, arXiv:1706.04599.
[57] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevB.98.060301 for analysis of the interpretability and robustness of the network predictions, comparison with a conventional order parameter for the MBL transition, and a detailed explanation of the training procedure.
[58] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, J. Mach. Learn. Res. **15**, 1929 (2014).
[59] D. P. Kingma and J. L. Ba, arXiv:1412.6980.
[60] A. Pal and D. A. Huse, Phys. Rev. B **82**, 174411 (2010).
[61] D. J. Luitz, N. Laflorencie, and F. Alet, Phys. Rev. B **91**, 081103 (2015).
[62] M. Serbyn, Z. Papić, and D. A. Abanin, Phys. Rev. X **5**, 041047 (2015).
[63] S. Bera, H. Schomerus, F. Heidrich-Meisner, and J. H. Bardarson, Phys. Rev. Lett. **115**, 046603 (2015).
[64] S. Iyer, V. Oganesyan, G. Refael, and D. A. Huse, Phys. Rev. B **87**, 134202 (2013).
[65] S. Gopalakrishnan, M. Knap, and E. Demler, Phys. Rev. B **94**, 094201 (2016).
[66] M. Schreiber, S. S. Hodgman, P. Bordia, H. P. Luschen, M. H. Fischer, R. Vosk, E. Altman, U. Schneider, and I. Bloch, Science **349**, 842 (2015).
[67] D. J. Luitz, N. Laflorencie, and F. Alet, Phys. Rev. B **93**, 060201 (2016).
[68] R. Gupta and M. J. Biercuk, Phys. Rev. Appl. **9**, 064042 (2018).
[69] M. Abadi *et al.*, arXiv:1603.04467.
[70] F. Chollet *et al.*, KERAS, https://keras.io/.